

発表概要

Ruby 処理系へのスナップショット GC の実装

相川 光^{†1} 笹田 耕一^{†1} 本位田 真一^{†1,†2}

オブジェクト指向スクリプト言語 Ruby はガーベッジコレクション (以下, GC) を備えており, プログラマがメモリ管理をする必要がないという特長を持つ. しかし, 現在広く利用されている Ruby 処理系は Stop The World GC を採用しており, 場合によってはアプリケーションの実行が長く中断されてしまうという問題がある. これを解決するため, 一連の GC 処理を細かく分割し, アプリケーションと並行して少しずつ実行するインクリメンタル GC という手法が考案されている. そこで, 我々は GC によるアプリケーションの停止時間を低減させるため, インクリメンタル GC の 1 つであるスナップショット GC を Ruby 処理系へ実装した. スナップショット GC にはオブジェクト間の参照関係を維持するための処理 (ライトバリア) が必要となるが, C 言語による拡張ライブラリが開発可能な Ruby においては, これらにもライトバリアを挿入しなければならない. このため, 本研究では GC 時にメモリ領域を書き込みから保護しておくことでライトバリア挿入の漏れを検出するという方式を用いて, 効率的にスナップショット GC を実装した. 本発表では Ruby 処理系におけるスナップショット GC の実装と開発, および評価結果について述べる.

Implementation of Snapshot GC on the Ruby Interpreter

HIKARI AIKAWA,^{†1} KOICHI SASADA^{†1}
and SHINICHI HONIDEN^{†1,†2}

Ruby, an object-oriented script language, has a Garbage Collection (GC) so that programmers need not manage memory areas. However, the pause time due to GC is sometimes long because GC on widely used Ruby interpreter is Stop The World GC. In order to avoid this problem, incremental GC which runs in parallel with the main program has been proposed. To shorten the pause time of the application for Ruby, we implement snapshot GC which is one of incremental GCs on the Ruby interpreter. In order to implement snapshot GC, a write barrier operation that keeps relationships between two objects is required. Since programmers can develop extension libraries in Ruby using the C language, we need to insert write barriers in such libraries as well.

We implement snapshot GC efficiently by protecting memory areas from being overwritten during GC. In this presentation, we explain the implementation of snapshot GC on the Ruby interpreter, the development and the result of evaluation.

(平成 20 年 10 月 28 日発表)

^{†1} 東京大学大学院情報理工学系研究科

Graduate School of Information Science and Technology, The University of Tokyo

^{†2} 国立情報学研究所

National Institute of Informatics